Computers & Education 56 (2011) 916–922

Contents lists available at ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

A team building model for software engineering courses term projects

Yasar Guneri Sahin*

Izmir University of Economics, Faculty of Engineering and Computer Sciences, Department of Software Engineering, Sakarya Cad. No: 156, Balcova, Izmir, Turkey

ARTICLE INFO

Article history:
Received 31 August 2010
Received in revised form
26 October 2010
Accepted 4 November 2010

Keywords:
Architectures for educational technology system
Evaluation methodologies
Gender studies
Improving classroom teaching
Team building

ABSTRACT

This paper proposes a new model for team building, which enables teachers to build coherent teams rapidly and fairly for the term projects of software engineering courses. Moreover, the model can also be used to build teams for any type of project, if the team member candidates are students, or if they are inexperienced on a certain subject. The proposed model takes students' preferences and the teacher's considerations into account when a team building process is required for any type of course. In addition, this paper investigates how team building models (RandomM: teams are built with randomly selected students; TeacherM: teacher selects the members for each team; StudentsM: students build their own teams and the proposed model) affect team performance and how gender differences affect project activities and team performance. A three-year (five semesters) teaching experiment was performed with the participation of 248 male and 79 female university students and a total of 67 software project teams. Two different One-way ANOVA tests were applied on the experimental data, and the results indicated that the proposed model was better than RandomM, TeacherM and StudentsM models in terms of project grades, and the effect of gender differences on the teams' performance and project activities was negligible.

1. Introduction

Teams are a primary mechanism for accomplishing organizational work, especially for software projects (Faraj & Sproull, 2000), hence team building, team size and cooperation between team members are critical factors in providing a quality software project. Team building process has been regularly investigated over a long period, and a wide range of methods have been implemented to improve team cohesion and quality.

1.1. Literature review

The first contextual study in this field, called "Tuckman's model of team development" (Tuckman, 1965), proposed a four-stage model, consisting of form, storm, norm and perform sequence. These stages are regarded as the best and idealized forms (Buchanan & Huczynski, 1997). However, Rickards and Moger (2000) proposed a new contextual framework that modifies Tuckman's well known model. They stated that "although the stages of Tuckman model may have considerable face validity as a general sequence, empirical observations of specific teams reveal complexities that cannot be explained as a simple stage sequence". They focused on two questions: "what if the storm stage never ends" and "what is needed to exceed performance norm", and proposed to replace all stages with two barriers: weak behavioral barrier and strong performance barrier. They have proposed an effective model, and have successfully tested their hypothesis with projects teams of business graduates and multiple teams within multidimensional industrial organizations. In addition, a number of modified and complementary alternative models to Tuckman's original have been proposed (Hope et al., 2005; Mcgrew, Bilotta, & Deeney, 1999; Miller, 2003; Rickards & Moger, 2000; Tuckman, 2010; Tuckman & Jensen, 1977).

Furthermore, team coherency and its effects on team performance have also been studied by many researchers. Chung and Guninan investigated the relationship between team members' participative style and team performance in software development, and they stated that team size and the professional experience of team members moderate the relationship between participation and performance. The

* Tel.: +90 232 488 8173; fax: +90 232 488 8475.
E-mail address: yasar.sahin@ieu.edu.tr.

0360-1315/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2010.11.006

most important finding is that participation is significantly related to team performance in teams with inexperienced members (Chung & Guinan, 1994). In addition, there are many other studies about factors that affect team performance, such as the problems caused by the distribution of projects to teams (in terms of workloads and size of the projects), difficulties in project management and collaboration problems within groups (Beck, 2008; Favela & Pena-Mora, 2001; Humphrey, Musson, & Salazar, 2008; Jenkins, 2008; Mead, 2009; Mitchell & Delaney, 2004; Pereira, Cerpa, Verner, Rivas, & Procaccino, 2008; Robillard & Robillard, 2000).

Bardach (2004) offered a solution to these problems, his method, "elimination auction", presents a computer program for matching students to particular projects. In this approach, students are asked to vote or bid for the projects from a project pool according to several rules. Finally, project groups (teams) are assembled by students with an interest in the particular project subjects. Although the method is ingenious clever, it is a top-down approach: first, the subject is decided, and then subject coherent teams are built according to the project subject. However, there are a number of disadvantages to this approach: well-adjusted teams may not be built, and project subjects have to be defined in advance. Thus, either a very limited project pool is offered to students, or they cannot freely select any of the real life project subjects they may be interested in. However, in a bottom-up approach, teams are built first, and then a subject is decided upon by team members. Thus, this approach seems to be more advantageous in terms of project subjects.

1.2. Motivation

While these studies state some kind of team problems, they propose several solutions to team building and team management problems for industrial and business projects. However, in education, especially in software engineering education, students as team member candidates are inexperienced, and this software project might be their first encounter with project issues. Students can be sensitive, and they need to be guided very carefully through their first experience in project tasks and management. In addition to students' project skills, group participation and relationships are crucial and need to be taken into account during the team building process.

Inexperience is not the only problem in building a new team for software engineering (SE) courses' term projects. The list below highlights a number of limitations and restrictions evident in the process:

1. Time limitation: there is very limited time for building the project teams. A semester is about 15 weeks, including exams and presentations; hence team building processes should be completed in a very short time. When the teacher lets the students build their project teams, the students cannot finalize building the teams in two weeks because of problems caused by relationships between students, emotionality, avoiding responsibility, etc. These delays result in very limited time to complete the project work. Therefore, team performance and project quality may suffer.
2. Behavioral tendency problems: since all team member candidates are students, they are likely to behave in ways considered unprofessional, such as gravitating toward their best friends or academically successful students. Empirical observations reveal that sometimes, students can be unwilling to be in the same project team with certain students because of hidden conflicts between them, but they may not express this directly, causing unnecessary tension or strife. This situation decreases the motivation and total effectiveness of the team.
3. Limited project subjects: forcing students (teams) to select a project subject from a limited project pool may narrow the students' visions. Therefore, a project team should be allowed to freely select a project subject from the real world, not only from a project pool. Furthermore, it is important that they discuss their availability and interests, such as interviewing and negotiating abilities, travel restrictions etc, before deciding what the proper project can be. This process is crucial to improving project skills and creativity.

In this paper, four team building models – *RandomM (*teams are built with the randomly selected students); *TeacherM (*teacher selects the members for each team); *StudentsM (*students build their own teams); and *ProposedM (*the proposed model) are examined. When the restrictions and limitations listed above are considered, *RandomM*, *TeacherM* and *StudentsM* models appear not to be the best choices for building project teams for SE courses' term projects.

This paper proposes a bottom-up, straightforward algorithm that is a combination of the improved model of Gale & Shapley and Prim's minimum spanning tree algorithm to solve team building problems for SE courses (Gale & Shapley, 1962; Prim, 1957). Gale and Shapley proposed a model to solve roommate matching problems (similar to team building for a software project). In addition, there is a recent proposal to solve roommate matching problems by Morrill (2010); however, he focused only on 2N students into N roommates pairing. Although these coupling methods need improvements in order to be used in the problem of N students pairing into M teams, their approaches are very useful for initial pairing of students into subgroups.

The model proposed in this paper is an improvement of roommate coupling method. By taking into account students' preferences, it improves team collaboration and team success, and limits team problems.

2. Team building for term projects

In this study, four different team building models –*RandomM; TeacherM; StudentsM; ProposedM*- are examined as previously mentioned for term projects of software engineering courses.

2.1. Team building models

RandomM: this model is based on random selection of *n* (team size) students from a student list for each team. The teacher makes a random selection algorithm to build teams at the beginning of the semester using a variety of options, ranging from maximum team size to gender (i.e. the teacher may decide to include at least one female in each team). If there is an undesired structure in any team, the teacher finalizes manually. This is the fastest model, however it is unfair, and the one that least considers the students as individuals.